



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-582
TITLE	RANDOM NUMBER GENERATOR ADAPTED FOR 8K FORTRAN/SABR
AUTHOR	Adapted by: W. F. Haygood, Jr.
COMPANY	3953 Starbrook Road Chesapeake, Virginia
DATE	October 7, 1972
SOURCE LANGUAGE	8K SABR

### ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.





## SUMMARY

This is a Random Number Generator adapted from DECUS 5-25 and designed for use specifically with DEC/s 8K FORTRAN and 8K SABR programming systems. The actual generator portion of the coding achieves results identical to DECUS 5-25.

The range  $-1 < x < +1$  has been chosen as the output of the generator. P. T. Brady's generator (DECUS 5-25) outputs octal numbers in the range 0000 through 7777. The generator incorporated herein likewise produces numbers in this range. The number is then floated and placed in the floating point accumulator (FAC) where it is divided by 2048 to put the final result in the range  $-1 < x < +1$ . If random integers are desired, the result can be multiplied by a suitable constant.

The generator is initialized to the value 0000 0000 0001<sub>8</sub> when it is first loaded. The user may initialize the generator to any desired value at any time by using a CALL to RANDI as explained below. Initializing the generator at the beginning of each programming run will eliminate having to call it a number of times to advance it to a state previously attained.

## USAGE

To initialize the generator under program control, use the following:  
for 8K FORTRAN:

```
CALL RANDI(J,K,L)
```

where J, K, and L are the decimal equivalents of the desired 36-bit initializing number (high, middle, and low order parts respectively). For example: if at the end of a program run, RANDOM (explained on page 2) contained the value 0200 1000 2000<sub>8</sub>, then the following FORTRAN statement would re-establish this value at the beginning of the next programming run:

```
CALL RANDI(128, 512, 1024)
```

for 8K SABR:

```
CALL 3, RANDI  
ARG J  
ARG K  
ARG L
```

where J, K, and L are the addresses of the high, middle, and low order parts of the desired 36-bit initializing number. For example: if at the end of a program run, RANDOM (explained on page 2) contained the value 0200 1000 2000<sub>8</sub>, then the following SABR coding would re-establish this value at the beginning of the next programming run:

```
CALL 3, RANDI  
ARG (200  
ARG (1000  
ARG (2000
```





## USAGE (continued)

It may, perhaps, be even easier to initialize the generator by manual means. When the 8K Linking Loader loads the generator, it will store the address call RANDOM. The location of RANDOM will be printed if a storage map is requested from the Linking Loader. RANDOM is the first word (high order word) of the 36-bit initializing number. To initialize the generator manually, proceed as follows:

- 1) Find the address of RANDOM from the Linking Loader Storage Map.
- 2) Set the Instruction Field switches and the Switch Register to the address of RANDOM. Press LOAD ADDRESS.
- 3) Set the Switch Register to the desired value of the high order word. Press DEPOSIT.
- 4) Set the Switch Register to the desired value of the middle order word. Press DEPOSIT.
- 5) Set the Switch Register to the desired value of the low order word. Press DEPOSIT.

Using the above manual method eliminates the necessity of programming the necessary instructions into the program. RANDI need not be called at all using this method.

To call the generator, use the following call:  
for 8K FORTRAN:

X=RANDU(I)

where (I) can be any value. This value is ignored by the generator, so normally just (0) would be used thus:

X=RANDU(0)

Of course, the Random Number Generator call may be nested with other calls as in the following statement:

X=SQRT(RANDU(0))

for 8K SABR:

CALL 1, RANDU  
ARG 0

To obtain random integers from the generator, the generator output may be multiplied by a suitable constant. For example:

for 8K FORTRAN:

I=RANDU(0)\*CONST

where CONST is the constant. Note that the constant must be floating point since the generator output is less than 1 in magnitude.

for 8K SABR:

CALL 1, RANDU  
ARG 0  
CALL 1, FMP  
ARG CONST

The remainder of this documentation consists of the SABR second pass listing, symbol table, and a sample FORTRAN program showing the output of this generator for 320 random numbers.





```

//// // // //
/
/A RANDOM NUMBER GENERATOR FOR
/DEC'S 8K FORTRAN/SABR SYSTEM
/
/THE ACTUAL GENERATOR IS DECUS 5-25
/
/ADAPTED FOR 8K FORTRAN/SABR BY
/W. F. HAYGOOD JR., CWO-3, USCG
/
/OCTOBER 7, 1972
/
/THE ACTUAL CODING CONSISTS OF 2 PARTS:
/      1) A SUBROUTINE WHICH INITIALIZES
/          THE RANDOM NUMBER GENERATOR
/      2) A FUNCTION WHICH IS THE
/          GENERATOR ITSELF
/
/THE GENERATOR MAY BE INITIALIZED TO ANY DESIRED
/36-BIT VALUE BY MEANS OF THE FOLLOWING CALL:
/      CALL RANDI(J,K,L)      /FOR FORTRAN
/          WHERE J, K, AND L ARE THE DECIMAL
/          EQUIVALENTS OF THE HIGH, MIDDLE,
/          AND LOW ORDER WORDS RESPECTIVELY.
/
/      OR
/      CALL 3,RANDI      /FOR SABR
/      ARG J
/      ARG K
/      ARG L
/
/          WHERE J, K, AND L ARE THE ADDRESSES
/          OF THE HIGH, MIDDLE, AND LOW ORDER
/          PARTS OF THE DESIRED 36-BIT STARTING
/          NUMBER.
/
/FOR EXAMPLE, THE FORTRAN CALL:
/      CALL RANDI(128, 512, 1024)
/          WOULD SET THE GENERATOR TO THE OCTAL
/          VALUE: 0200 1000 2000 (36 BITS USED)
/
//// // // //

```





```

//// // // //
/
/INITIALIZING THE GENERATOR WOULD ELIMINATE HAVING TO
/CALL IT A NUMBER OF TIMES TO MAKE IT "RANDOM" AGAIN
/AFTER LOADING AND USING IT THE FIRST TIME. IT ALSO
/ALLOWS THE REPETITION OF "RANDOM" INPUT DATA FOR USE
/IN TESTING PROGRAMS.
/
/THE RANDOM NUMBER GENERATOR MAY BE CALLED AT ANY TIME
/WHETHER IT HAS BEEN INITIALIZED OR NOT. USE THE
/FOLLOWING FORTRAN STATEMENT TO CALL IT:
/      X=RANDU(I)
/      WHERE I CAN BE ANY VALUE (USUALLY 0).
/
/SINCE RANDU RETURNS NUMBERS IN THE RANGE  $-1 < X < +1$ ,
/A SUITABLE CONSTANT MULTIPLIER CAN BE USED FOR
/PRODUCING INTEGER NUMBERS THUS:
/      J=RANDU(0)*2048
/
//// // // //

```





		ENTRY RANDI	/GENERATOR INITIALIZER
		ENTRY RANDU	/GENERATOR ITSELF
		ENTRY RANDOM	
		DUMMY J	
		DUMMY K	
		DUMMY L	
		RANDOM,	
0200	0000	HIGH,	0
0201	0000	MID,	0
0202	0001	LOW,	1
		J,	
		K,	
		L,	BLOCK 2
0203	0000		
0204	0000		
0205	0000	RANDI,	BLOCK 2
0206	0000		
0207	4067		
0210	0205 01	TAD I RANDI	/GET ADDR OF J
0211	1407		
0212	2206	INC RANDI#	
0213	3203	DCA J	
0214	4067	TAD I RANDI	
0215	0205 01		
0216	1407		
0217	2206	INC RANDI#	
0220	3204	DCA J#	
0221	4067	TAD I J	/GET VALUE OF J
0222	0203 01		
0223	1407		
0224	3200	DCA HIGH	/HIGH ORDER INITIALIZED





0225	4067				
0226	0205	01	TAD I RANDI	/GET ADDR OF K	
0227	1407				
0230	2206		INC RANDI#		
0231	3203		DCA K		
0232	4067		TAD I RANDI		
0233	0205	01			
0234	1407				
0235	2206		INC RANDI#		
0236	3204		DCA K#		
0237	4067		TAD I K	/GET VALUE OF K	
0240	0203	01			
0241	1407				
0242	3201		DCA MID	/MIDDLE ORDER INITIALIZED	
0243	4067		TAD I RANDI	/GET ADDR OF L	
0244	0205	01			
0245	1407				
0246	2206		INC RANDI#		
0247	3203		DCA L		
0250	4067		TAD I RANDI		
0251	0205	01			
0252	1407				
0253	2206		INC RANDI#		
0254	3204		DCA L#		
0255	4067		TAD I L	/GET VALUE OF L	
0256	0203	01			
0257	1407				
0260	3202		DCA LOW	/LOW ORDER INITIALIZED	
0261	4040		RETRN RANDI		
0262	0001	06			





	////	////	////	////
0263	0000	RANDU,	BLOCK 2	
0264	0000			
0265	1202		TAD LOW	/BEGIN DECUS 5-25
0266	7004		RAL	
0267	0377		AND (7400	
0270	3361		DCA TEMP2	
0271	1201		TAD MID	
0272	0376		AND (177	
0273	1361		TAD TEMP2	
0274	7006		RTL	
0275	7006		RTL	
0276	7004		RAL	
0277	3357		DCA TEMP	
0300	1202		TAD LOW	
0301	0376		AND (177	
0302	7006		RTL	
0303	7006		RTL	
0304	7004		RAL	
0305	1201		TAD MID	
0306	3360		DCA TEMP1	
0307	7420		SNL	
0310	2357		INC TEMP	
0311	7000		NOP	
0312	1202		TAD LOW	
0313	7104		CLL RAL	
0314	7420		SNL	
0315	2360		INC TEMP1	
0316	7410		SKP	
0317	2357		INC TEMP	
0320	7000		NOP	
0321	7100		CLL	
0322	1202		TAD LOW	
0323	3202		DCA LOW	
	/	////	////	////





0324	1201		TAD MID	
0325	7004		RAL	
0326	7420		SNL	
0327	2357		INC TEMP	
0330	7000		NOP	
0331	7100		CLL	
0332	1360		TAD TEMP1	
0333	3201		DCA MID	
0334	1200		TAD HIGH	
0335	7004		RAL	
0336	1200		TAD HIGH	
0337	1357		TAD TEMP	
0340	3200		DCA HIGH	/END OF DECUS 5-25
0341	2264		INC RANDU#	/SET RETURN POINTER
0342	2264		INC RANDU#	
0343	1200		TAD HIGH	
0344	4033		CALL 0, FLOT	/PUT RANDOM NUMBER IN FAC
0345	0004 06			
0346	4033		CALL 1, FDV	/PUT IN RANGE -1 TO +1
0347	0105 06			
0350	6201 05		ARG C	
0351	0354 01			
0352	4040		RETRN RANDU	
0353	0002 06			
0354	2144	C,	2144	/C=2048(10)
0355	0000		0	
0356	0000		0	
0357	0000	TEMP,	0	
0360	0000	TEMP1,	0	
0361	0000	TEMP2,	0	
		/		
0376	0177			
0377	7400			

END



C	0354
FDV	0000EXT
FLOT	0000EXT
HIGH	0200
J	0203
K	0203
L	0203
LOW	0202
MID	0201
RANDI	0205EXT
RANDOM	0200EXT
RANDU	0263EXT
TEMP	0357
TEMP1	0360
TEMP2	0361





```

C - SAMPLE RANDOM NUMBER GENERATOR OUTPUT
  CALL RANDI(128, 512, 1024)
  DO 20 J=1, 40
    DO 10 I=1, 8
      X=RANDU(0)
10    WRITE (1,30) X,
20    WRITE (1,30)
      STOP
30    FORMAT (2H ,F7.4)
      END

```

.1924	.6060	-.1147	-.1279	.2461	.6421	-.3813	-.0537
-.9062	-.9404	.4961	-.5459	.2422	.3799	.0820	-.9131
-.2344	.8247	-.9590	.8389	-.3535	.3428	-.7793	.2524
.5122	.8149	.2622	.2529	-.8613	.5693	-.8496	-.2080
.3809	.1709	-.4199	-.0439	-.5015	-.6011	.8892	.7583
.5303	.3691	-.5747	-.7583	.6050	.4683	-.6528	-.1177
-.8481	-.0156	-.4795	-.7227	-.0381	.2886	.0581	-.2349
.0508	.4307	.1094	.7935	-.2397	-.5654	.7461	-.4209
.7422	.2549	.8320	.7119	.7651	.1982	.2871	-.0483
-.8901	-.8916	.6431	-.1045	-.4331	.3540	.0039	.8525
-.9375	.7158	.7148	-.1392	.7129	-.4580	.8179	-.9561
.8867	-.0610	-.3657	.3687	-.5146	-.3911	.2686	-.8540
.4429	.3569	.1382	-.3701	.5195	.4619	.0781	.3252
-.7695	.4697	-.2739	.1440	-.6880	.5894	-.2896	.9722
.4214	-.2075	.9453	-.4482	.7847	.7573	-.5352	-.0146
.7109	.4111	.0513	.6196	-.7612	-.1309	.0498	-.5103
.4741	-.5493	.4199	-.5244	-.9434	-.9268	.9116	-.1748
.7305	-.0303	-.7739	-.3560	.8120	.0898	-.7881	.4766
-.0659	-.6694	.5591	-.6069	-.6909	-.6689	.1855	-.8530
-.8042	.8657	.4150	.7109	.5127	.6914	-.4834	.8901
-.3247	.0542	-.7700	.9058	.3472	-.0547	.5288	-.3228
-.7134	.6377	.2305	-.3428	-.1489	.2061	.5601	-.4790
.0674	.7305	-.2417	-.0093	.1021	.7095	-.6792	-.4468
-.5854	.5215	.3813	-.3906	.2075	.7759	.7695	-.3511
.9517	.8828	.7153	.3618	-.2842	-.9492	.8447	-.3755
.1279	.1621	-.1963	-.6226	.0122	-.3105	.0107	.8730
-.8765	.8989	-.7358	-.4912	-.3437	.3716	-.6929	.5117
-.7104	-.8530	-.7422	-.7607	.0977	-.5537	-.2187	-.3159
.0571	-.7998	.6680	-.7808	-.7129	.7646	.9863	-.9502
-.5957	.9912	-.7090	.8384	-.6050	.8384	.4575	-.7866
-.8555	-.0386	-.5483	-.9292	-.6572	.4316	.4873	-.9473
-.0869	.0176	.8701	-.9238	.6084	-.0215	.3779	.4741
-.5728	.3101	.9976	-.7915	.2554	.6694	-.2993	.1924
-.1685	-.7275	-.8652	-.6299	-.0103	-.3774	-.1904	.2695
-.6860	-.5293	.9795	.6543	-.9077	.6802	.2324	-.7124
-.3828	.1294	.2056	.0825	.6274	-.9644	.5488	-.0132
.9653	-.0757	.8403	-.2632	.8403	-.5757	.9653	.9868
-.7847	.4243	-.4097	-.2632	.0903	.9243	.7153	-.0132
-.5342	.9258	.3447	-.2505	-.6216	.5391	.8096	.0195
.8135	.7187	.9736	-.6128	-.4580	.7793	.7803	-.3188



